



マイクロサービス アーキテクチャと データベース

SIOS Technology, Inc.

Innovative Solution Business Planning Dept.

Tatsuhiko Murata

講師紹介

- 講師紹介

Name : 村田 龍洋

Company : サイオステクノロジー株式会社

Dept : イノベーティブソリューション事業企画部

- Task

2002年 サイオステクノロジー（当時のテンアート二）に入社

ハードウェア部門、Linux/OSSサポート部門を経て、現在の事業企画へ

AWSパートナー契約やNGINXパートナー契約を担当、OSSよろず相談室の企画・運営を経て

現在はエンタープライズ向けのOSSサブスクリプション製品

（EnterpriseDB, Red Hat, NGINX）の事業企画を担当

IaC活用研究会 運営メンバー

マイクロサービスが求められる背景

デジタルへの変革が求められるビジネス

現在、ビジネスの変化は非常に早くあらゆる業種で起きています。
そして、企業の成長にはデジタルコンテンツが欠かせません。

市場へいかに早く革新的なサービスを提供するか、スピードの重要度が高まり、その流れに対応できるかが勝負の分かれ目になります。
しかし、多くの企業ではこの変革を起こすことができずにいます。

デジタル・トランスフォーメーション（DX）というキーワードで取り組みを始める企業が増加中

実店舗

Webサイト

スマートフォンアプリ

小売のビジネスを例に見ると、これまで実店舗での販売が中心であったものが、Webサイト、そしてスマートフォンアプリへと変化してきている。

大きな変化が起きたモノとコト

商品販売

新聞紙、雑誌

テレビ、CD、DVD

製造・生産

交通・移動手段

銀行、証券



**あらゆる業種で
ビジネスがデジタルへと
切り替わっています。**

破壊的イノベーション

Disruptive technology :

従来の価値基準のもとではむしろ性能を低下させるが、新しい価値基準の下では従来製品よりも優れた特長を持つ新技術

Disruptive technologyの主な例

- iPhone
- Facebook
- Uber

破壊的サービスの多くはスマートフォンが媒体に



5G ネットワークの破壊力

第5世代移動通信システムの特徴（4Gとの比較）

- 「高速大容量」 : 1Gbpsから10Gbps
- 「低遅延」 : 10msecから1msec
- 「同時多接続」 : 10万デバイス/km²から100万デバイス/km²

音声通信からデータ通信、
そして時代は**サービスコンテンツデリバリー**の時代へ

マイクロサービスのはじまり

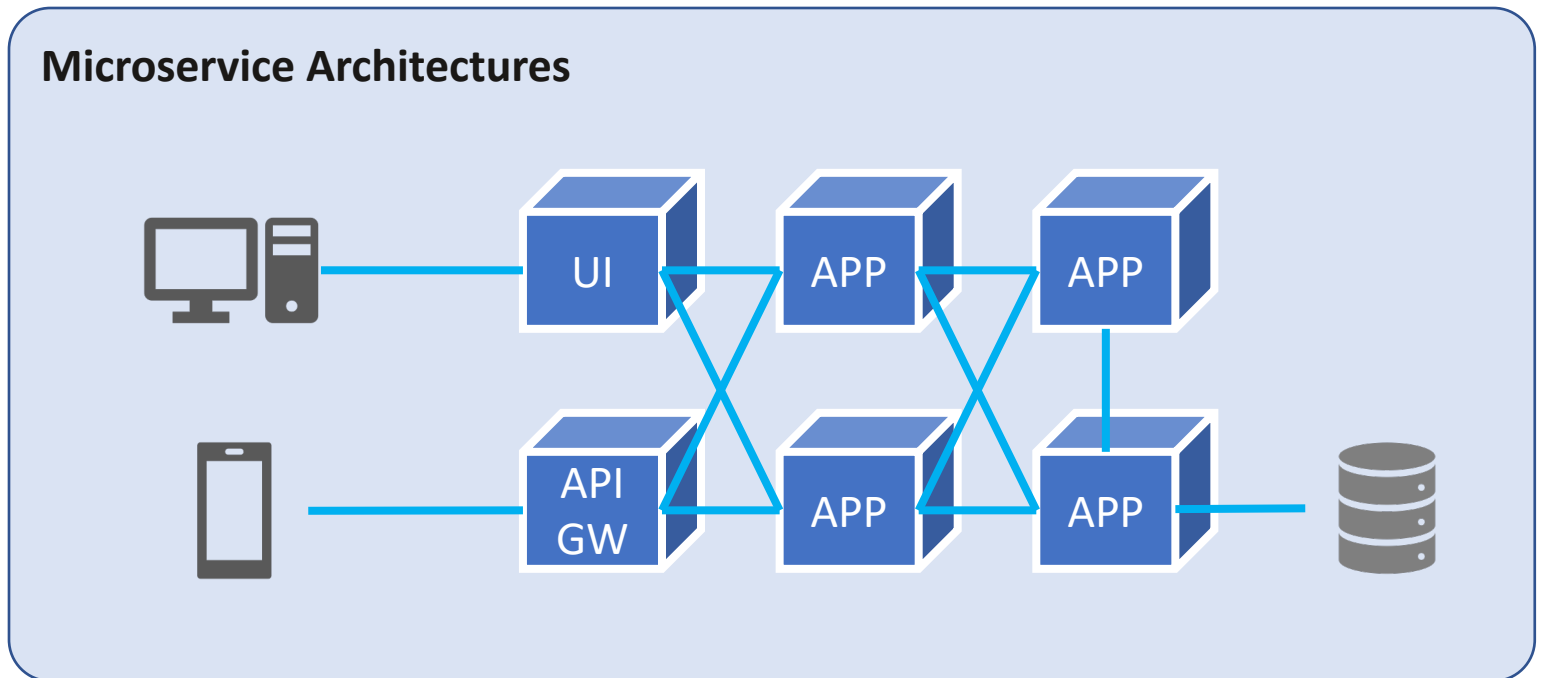
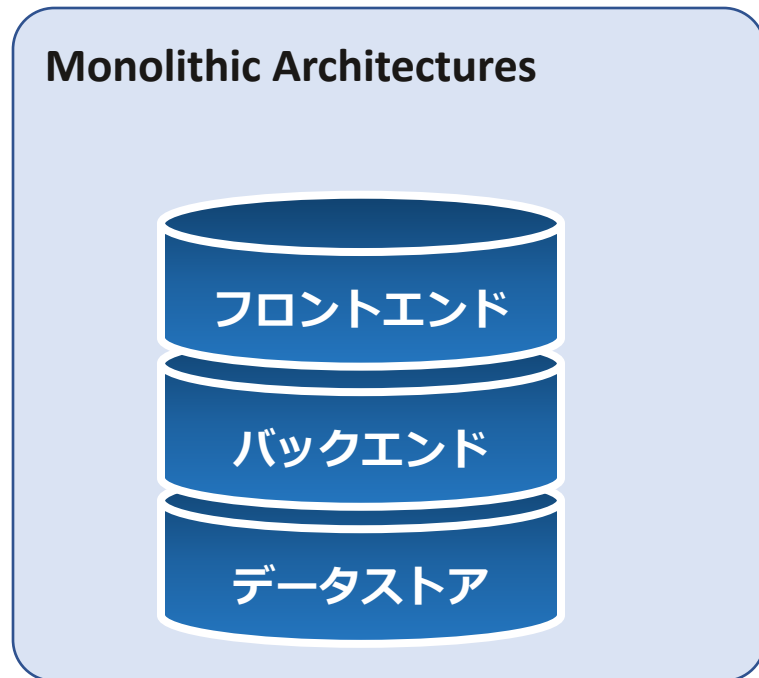
マーチン・ファウラー氏らが2014年に公開した「Microservices」という記事より世に広まったと言われています。

マイクロサービスは、これまで主流であった**モノリス**なアプリケーションとは異なる手法で、機能ごとに最小のアプリケーションを作成し、疎結合で繋ぐのが特徴です。

海外ではAmazonや、Netflixなどの企業で多く採用されており、日本でもLINEやクックパッド、Gunosyなどのサービスで利用されています。

モノリスとマイクロサービス

- 3層構造で全ての機能を1つのアプリケーションとして作成するモノリスアプリケーションに対して、マイクロサービスは機能ごとにアプリケーションを作成する。



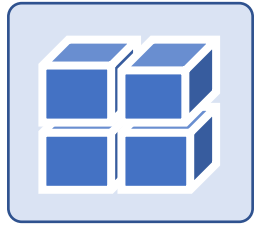
Monolithic Architecturesの特徴



全ての機能を1つのパッケージに
開発、テストが全体を通して行うため、確実な手法

- アプリケーションがパッケージ化され、ソフトウェアベースで展開
- 拡張する場合、サーバーにアプリケーションをコピーして展開
- 拡張の度に肥大化するアプリケーション
- 障害発生時の影響は全体に及び切り分けの難易度はコード量に比例
- バグ修正や新機能追加時の影響範囲はコード量に比例して増加

Microservice Architecturesの特徴



機能ごとにアプリケーション最小化開発し、
アプリケーション同士をREST APIなど疎結合で繋ぐ

- アプリケーション開発の高速化と単純化
- サービスに適した開発手法、言語・技術を適用
- アプリケーションの独立性を持たせるため、各サービス毎にデータベースを持たせる必要がある。
- アップデート時、他のサービスへの影響小
- スケールアウトがサービス毎に独立して可能
- サービス全体を通じてのテストシナリオは複雑化

マイクロサービスの利点

■技術の多様性

サービスごとに最適な言語を利用して各サービスの開発が行える

■個別デプロイ

変更をかけたいときは、システム全体ではなく、小さなサービスごとに変更をかけられるため、影響範囲が限定的

■開発効率

小さなサービスで開発単位を進めるため、ビルドやテストの期間が短くなり開発効率が向上

■障害時の影響範囲

マイクロサービスでは障害時の影響範囲が限られ、原因の突き止めが比較的容易

マイクロサービスの課題

■データの一貫性

特にトランザクション処理系はマイクロサービスでは不向きとされています。

■分散化するサービス

サービスが細かくいくつも立ち上がるため、管理対象が増加し、煩雑化する可能性がある

■マインドチェンジへの抵抗

開発と運用がそれぞれ連携する意識が必要になるため、マインドチェンジが求められる。

PMやアーキテクトにはこの連携も意識した設計が求められる。

マイクロサービスで起こる変化

マイクロサービスを導入することで、様々なところで変化を求められる事になります。一例を上げただけでも以下のような変化が想定出来ます。

- 開発手法と開発環境
- PM、アーキテクトに求められる条件
- 運用の範囲と運用方法
- システム毎のライフサイクル

※マイクロサービスはあくまでも手段のため、Dev/Opsという言葉も使われるように、運用と開発が一体となりサービスの開発・運用を行うというマインドの変化がなければ導入は失敗に終わる可能性もある。

マイクロサービスへ従来の手法を持ち込むと？

■ 運用の煩雑化

- ▶ 数百、数千、時には数万のリソースをどのように運用するのか？
短縮されるライフサイクルを人の手だけで運用ができるのか？

■ 運用コストの増加

- ▶ サービスの拡大と共にシステムにかかるコストが増加
そして、人員の増強も求められ、人件費が増加

■ マネージメント層もしくは技術者の抵抗

- ▶ 手順書通り、仕様書通りの運用が安全
- ▶ 新たな技術習得への抵抗、誰が？どこから？どのように？の旗振り役が不在

従来の考え方のままでは評価・検証フェーズから進まないプロジェクトとなりやすい

マイクロサービスの導入に求められること

■クラウドやコンテナ技術、ツール類の選択

- ▶数百、数千、数万とサービス拡大と共にスケールするリソースに対応できるクラウド技術、コンテナ技術、自動化技術の選択は欠かせません。
また、開発もこの様なスケールするインフラを前提にアプリケーション開発を行う必要がある。

■技術範囲の拡大

- ▶クラウド、コンテナ、自動化、APIなど、新たな技術への対応が必要
常にアンテナを高く張り、最新技術の収集と習得が求められる

■体質改革、組織改革、企業文化改革

- ▶問題が発生しないための運用では新たな取組の足かせになる
日本企業の丸投げ体質もマイクロサービスでは大きな障害になります。
SRE (Site Reliability Engineering) 組織を作り改革する企業も増加している。

マイクロサービスでのデータベースの課題

大規模なマイクロサービスでデータベースを構築・実行・メンテナンスを行うことは非常に難易度の高い業務になります。

- アプリケーション毎に独立した設計が必要
- スケールアップ、スケールアウトを前提
- リソースの競合が発生しないこと
- 障害に対する耐性



サイオステクノロジー株式会社
イノベーティブソリューション事業企画部

EDB製品に関するご相談は下記フォームよりお気軽にご連絡ください。

<https://sios.jp/products/oss/postgres/contact/>